

## CLAIMS

What is claimed is:

- 1           1.     A method comprising:  
2           determining a processor state of a processor upon expiration of a system  
3 management interrupt (SMI) timer, the processor state being one of an operational state  
4 and a low power state;  
5           loading the SMI timer with a timer value based on the processor state, the timer  
6 value being one of a first value and a second value; and  
7           transitioning the processor to one of the operational state and the low power state  
8 according to the processor state.
- 1           2.     The method of claim 1 wherein loading the SMI timer comprises:  
2           loading the SMI timer with the first value if the processor state is the operational  
3 state; and  
4           loading the SMI timer with the second value if the processor state is the low power  
5 state.
- 1           3.     The method of claim 2 wherein transitioning comprises:  
2           transitioning the processor to the operational state if the processor state is the low  
3 power state; and  
4           transitioning the processor to the reduced power state if the processor state is the  
5 operational state.
- 1           4.     The method of claim 1 further comprising:  
2           disabling the SMI timer if throttling is disabled; and  
3           enabling the SMI timer if throttling is enabled.
- 1           5.     The method of claim 1 further comprising:  
2           generating a SMI access to a throttling state in response to an input/output (I/O)  
3 trap;

4 updating the throttling state if the access is a write; and  
 5 returning the throttling state if the access is a read.

1 6. The method of claim 5 wherein generating the SMI access comprises:  
 2 reporting the throttling state at a dummy address; and  
 3 generating the I/O trap by an SMI handler using the dummy address, the I/O trap  
 4 generating the SMI access.

1 7. The method of claim 6 wherein generating the I/O trap comprises:  
 2 invoking the SMI handler to trap on the dummy address;  
 3 booting a power management OS;  
 4 loading the dummy address by the power management OS;  
 5 accessing the throttling state at the dummy address by the power management OS;  
 6 and  
 7 generating the I/O trap by a chipset in response to accessing the throttling state by  
 8 the power management OS.

1 8. The method of claim 4 wherein booting the power management OS  
 2 comprises:  
 3 booting an Advanced Configuration and Power Interface (ACPI) OS.

1 9. The method of claim 3 wherein transitioning the processor to the low power  
 2 state comprises:  
 3 transitioning the processor to one of a first power state, a second power state, a  
 4 third power state, and a sleep state.

1 10. The method of claim 1 wherein loading the SMI timer comprises:  
 2 loading the SMI timer in a chipset.

1 11. A computer program product comprises:  
 2 a machine useable medium having computer program code embedded therein, the  
 3 computer program product having:

4 computer readable program code to determine a processor state of a  
 5 processor upon expiration of a system management interrupt (SMI) timer, the  
 6 processor state being one of an operational state and a low power state;  
 7 computer readable program code to load the SMI timer with a timer value  
 8 based on the processor state, the timer value being one of a first value and a second  
 9 value; and  
 10 computer readable program code to transition the processor to one of the  
 11 operational state and the low power state according to the processor state.

1 12. The computer program product of claim 11 wherein the computer readable  
 2 program code to load the SMI timer comprises:  
 3 computer readable program code to load the SMI timer with the first value if the  
 4 processor state is the operational state; and  
 5 computer readable program code to load the SMI timer with the second value if the  
 6 processor state is the low power state.

1 13. The computer program product of claim 12 wherein the computer readable  
 2 program code to transition comprises:  
 3 computer readable program code to transition the processor to the operational state  
 4 if the processor state is the low power state; and  
 5 computer readable program code to transition the processor to the reduced power  
 6 state if the processor state is the operational state.

1 14. The computer program product of claim 11 further comprising:  
 2 computer readable program code to disable the SMI timer if throttling is disabled;  
 3 and  
 4 computer readable program code to enable the SMI timer if throttling is enabled.

1 15. The computer program product of claim 11 further comprising:  
 2 computer readable program code to generate a SMI access to a throttling state in  
 3 response to an input/output (I/O) trap;  
 4 computer readable program code to update the throttling state if the access is a  
 5 write; and

6 computer readable program code to return the throttling state if the access is a read.

1 16. The computer program product of claim 15 wherein the computer readable  
2 program code to generate the SMI access comprises:

3 computer readable program code to report the throttling state at a dummy address;  
4 and

5 computer readable program code to generate the I/O trap by an SMI handler using  
6 the dummy address, the I/O trap generating the SMI access.

1 17. The computer program product of claim 16 wherein the computer readable  
2 program code to generate the I/O trap comprises:

3 computer readable program code to invoke the SMI handler to trap on the dummy  
4 address;

5 computer readable program code to boot a power management OS;

6 computer readable program code to load the dummy address by the power  
7 management OS;

8 computer readable program code to access the throttling state at the dummy address  
9 by the power management OS; and

10 computer readable program code to generate the I/O trap by a chipset in response  
11 to accessing the throttling state by the power management OS.

1 18. The computer program product of claim 4 wherein the computer readable  
2 program code to boot the power management OS comprises:

3 computer readable program code to boot an Advanced Configuration and Power  
4 Interface (ACPI) OS.

1 19. The computer program product of claim 3 wherein the computer readable  
2 program code to transition the processor to the low power state comprises:

3 computer readable program code to transition the processor to one of a first power  
4 state, a second power state, a third power state, and a sleep state.

1 20. The computer program product of claim 11 wherein the computer readable  
2 program code to load the SMI timer comprises:

3 computer readable program code to load the SMI timer in a chipset.

1 21. A system comprising:

2 a processor;

3 a memory coupled to the processor to store a throttling emulator, the throttling  
4 emulator, when executed, causing the processor to:

5 determine a processor state of the processor upon expiration of a  
6 system management interrupt (SMI) timer, the processor state being one of an  
7 operational state and a low power state;

8 load the SMI timer with a timer value based on the processor state,  
9 the timer value being one of a first value and a second value; and

10 transition the processor to one of the operational state and the low  
11 power state according to the processor state.

1 22. The system of claim 21 wherein the throttling emulator causing the  
2 processor to load causes the processor to:

3 load the SMI timer with the first value if the processor state is the operational state;  
4 and

5 load the SMI timer with the second value if the processor state is the low power  
6 state.

1 23. The system of claim 22 wherein the throttling emulator causing the  
2 processor to transition causes the processor to:

3 transition the processor to the operational state if the processor state is the low  
4 power state; and

5 transition the processor to the reduced power state if the processor state is the  
6 operational state.

1 24. The system of claim 21 wherein the throttling emulator, when executed,  
2 further causes the processor to:

3 disable the SMI timer if throttling is disabled; and

4 enable the SMI timer if throttling is enabled.

1           25.    The system of claim 21 wherein the throttling emulator further causes the  
2 processor to:  
3           generate a SMI access to a throttling state in response to an input/output (I/O) trap;  
4           update the throttling state if the access is a write; and  
5           return the throttling state if the access is a read.

1           26.    The system of claim 25 wherein the throttling emulator causing the  
2 processor to generate the SMI access causes the processor to:  
3           report the throttling state at a dummy address; and  
4           generate the I/O trap by an SMI handler using the dummy address, the I/O trap  
5 generating the SMI access.

1           27.    The system of claim 26 wherein the throttling emulator causing the  
2 processor to generate the I/O trap causes the processor to:  
3           invoke the SMI handler to trap on the dummy address;  
4           boot a power management OS;  
5           load the dummy address by the power management OS;  
6           access the throttling state at the dummy address by the power management OS; and  
7           generate the I/O trap by a chipset in response to accessing the throttling state by  
8 the power management OS.

1           28.    The system of claim 24 wherein the throttling emulator causing the  
2 processor to boot the power management OS causes the processor to:  
3           boot an Advanced Configuration and Power Interface (ACPI) OS.

1           29.    The system of claim 23 wherein the throttling emulator causing the  
2 processor to transition the processor to the low power state causes the processor to:  
3           transition the processor to one of a first power state, a second power state, a third  
4 power state, and a sleep state.

1           30.    The system of claim 31 wherein the throttling emulator causing the  
2 processor to load the SMI timer causes the processor to:

- 3           load the SMI timer in a chipset.